REMARKS

I. Summary of the Office Action and this Reply

Claims 1, 3-13, 16 and 18-30 are pending in the application. Claims 1, 3-13 stand rejected under 35 U.S.C. §103(a) as being obvious over U.S. Patent No. 6,530,075 to Beadle ("Beadle") in view of U.S. Patent No. 4,852,173 to Bahl et al. ("Bahl"). Claims 16 and 18-30 stand rejected under 35 U.S.C. §103(a) as obvious over Beadle in view of U.S. Patent Application No. 2004/0221272 to Wu ("Wu") and further in view of U.S. Patent No. 6,640,286 to Kawamoto et al. ("Kawamoto"). The Examiner raised objections as to claims 1 and 3 - 13.

In this Reply, claims 1 and 3-13 are amended to obviate the Examiner's objections. Claims 1 and 16 are amended for clarity.

Applicant does not acquiesce to the Examiner's characterizations of either the art of record or Applicant's subject matter recited in the pending claims. Further, Applicant is not acquiescing to the Examiner's statements as to the applicability of the prior art of record to the pending claims by filing this Reply. This Reply is intended to be a full and complete response to the Office Action dated June 4, 2008.

II. Brief Discussion of the Claimed Invention

An aspect of the claimed invention relates to annotating p-code methods with specific priority level hints, and enabling preferential processing of the p-code methods based upon the priority levels assigned. By assigning priority levels, a memory-constrained target environment that has insufficient memory to just-in-time (JIT) compile all p-code will devote the available memory to storing JIT compilation of portions of the p-code according to the order of priority assigned to those portions of the p-code. Thus, in a simplified example, if only one p-code method can be compiled (or stored in a cache) in view of limited resources, those limited resources will be devoted to compiling (or storing) the one p-code method having been annotated to have the highest priority; the limited resources will not be devoted to compiling (or storing) a lower prioritized p-code method before a higher prioritized p-code method. See Summary.

III. Brief Discussion of the Selected Art

A. Beadle

Beadle discloses just-in-time (JIT)/Compiler Java language extensions to enable field performance and serviceability. Specifically, Beadle provides a programmer with keyword extensions of the Java language to indicate which Java objects, classes, methods or code sections are to be just-in-time compiled. Thus, Beadle teaches that certain Java objects may be marked to be JIT compiled.

B. Bahl

Bahl discloses design and construction of a binary-tree system for language modeling that facilitates determining of a next event based upon available data, which is useful, for example, in the context of speech recognition to determine the next word likely to be spoken as a function of previously spoken words. Abstract. By constructing a decision tree having binary (true/false) questions at each node, one can traverse the tree by commencing at the root, answering the question at each node and following corresponding branches until a leaf is reached, the leaf representing the expected next word. See Col. 2, lines 1-32.

IV. Response to 103 Rejections

It is well-established that for a proper rejection under section 103, all claim limitations must be taught or suggested by the prior art.

The present invention specifically teaches enabling preferential processing performed as a function of hierarchically-related priority levels assigned to p-code methods in a p-code file. Neither Beadle, Bahl, nor Kawamoto discloses prioritizing p-code methods and enabling preferential processing of the p-code methods based on hierarchically-ordered priority levels, as claimed. Reconsideration and withdrawal of the rejection of claims 1, 3-13, 16 and 18-30 are requested respectfully.

A. Neither Beadle Nor Bahl Teaches Or Suggests Hierarchical Priority Level Hints Enabling Preferential Processing Of Annotated Methods In A Hierarchical Manner

Beadle teaches that certain Java objects may be marked to be compiled with a binary (true/false) marker. However, neither Beadle nor Bahl teaches or suggests any assigning of priority-level information to code sections to be compiled. Further, neither Beadle nor Bahl teaches or suggests annotating p-code methods with priority level hints collectively representing a hierarchical order, or providing any annotations enabling preferential processing of annotated methods in a hierarchical manner corresponding to the hierarchical order of the priority level hints. The claimed method allows a programmer to control the order and/or priority for a future compiling process, to ensure advantageous allocation of system resources during compiling. This is neither taught nor suggested by Beadle or Bahl. Instead, Beadle teaches only providing "FALSE" and "TRUE" markers to distinguish between portions intended to be JIT compiled and portions that are not to be JIT compiled. Such binary TRUE/FALSE markers are not the claimed hierarchically-related priority level hints; they do not collectively represent a hierarchical order, as required by claim 1. The Action acknowledges this distinction on page 4.

The Action asserts that Bahl discloses “a hierarchical manner corresponding to a hierarchical order” at col. 2, line 62 – col. 3, line 12. This section of Bahl states:

> Initially, all the data is at the root of the tree to be constructed. Commencing at the root, each node will be split into two descendent nodes until an entire binary tree has been created.
>
> The binary tree is constructed with reference to FIG. 1 by commencing at the root node 1, and splitting the root node into two subnodes 2,3 in order to minimize the average of the entropies of Y at each of the subnodes. The process continues by creating subnodes 4,5 and 6,7 from subnodes 2 and 3, respectively. The criteria used to determine whether to create the subnode is dependent upon only a single operation. Namely, finding a question which when applied to a node, minimizes the entropy of the resulting two subnodes. The question in the case of an unordered variable, such as in speech, is in the form "is X a member of the set S?", where X is one of the last N words spoken and set S is built by adding words to a set in the example of speech modelling. Entropy minimization reduces uncertainty of a decision.

This teaching of Bahl is entirely unrelated to the claimed invention. Rather, the claimed invention relates generally to (1) processing of p-code methods annotated with specific hierarchically-related priority levels, and processing the annotated methods in a hierarchical manner corresponding to a hierarchical order of their priority levels, or (2) preferentially retaining compiled p-code methods in a hierarchical manner corresponding to a hierarchical order of hierarchically-related priority level annotations, as recited with greater specificity in claims 1 and 16. By way of example, all (or many) p-code methods may be processed sequentially as sequential code, regardless of their respective priority annotations, and the compiled p-code methods may be retained in a limited-capacity cache according to a priority level, e.g. such that an earlier-compiled lower-priority p-code method is deleted from a cache and replaced with a later-compiled higher-priority p-code method, which is then retained in the cache despite subsequent processing of lower-priority p-code methods. See application, page 9, lines 5-27; page 12, lines 12-22.

Further, any vague teaching in Bahl of hierarchical relationships in the context of a binary tree having nodes, subnodes, a root and leaves as applied to pattern recognition to predict a next-spoken word as a function of context provided by previously-spoken words is entirely unrelated to the claimed invention. Any combination of the teachings of Bahl with the teachings of Beadle does not amount to the claimed invention. Specifically, neither Beadle nor Bahl, alone or in combination, teaches or suggests annotating p-code methods with <u>hierarchically-related priority level hints collectively representing a hierarchical order</u>, or <u>providing any annotations enabling preferential processing of annotated methods in a hierarchical manner corresponding to the hierarchical order of the priority level hints</u>. Bahl's teaching that nodes and subnodes in a tree are arranged hierarchically, even when coupled with the teachings of Beadle, do not amount to a teaching of annotating p-code methods with hierarchically-related priority level hints so that code can later be selectively JIT compiled in a hierarchical manner corresponding to the hierarchical order of the priority level hints, e.g. to better utilize system resources. For example, the teaching of Beadle involving marking code with TRUE (JIT compile) and FALSE (do not JIT compile) markers combined with the node/subnode teachings of Bahl does not amount to annotating code sections with hierarchically-related priority level hints (e.g., priority level 1, priority level 2, priority level 3, etc.) collectively representing a hierarchical order, so that the corresponding code sections can be selectively JIT complied in a hierarchical manner corresponding to the order of the priority level hints, e.g. with first preference being given to the p-code method(s) annotated with priority level 1, second preference being given to the p-code method(s) annotated with priority level 2, third preference being given to the p-code method(s) annotated with priority level 3, etc.

Further, the Examiner asserts that there is motivation to combine the references for a specific purpose, namely, as stated on page 4 of the Action, "to construct true/false decisions as a binary decision tree as well-known in the art as suggested by Bahl." This asserted reason for the combination is flawed. The claimed invention does not relate to true/false decisions as a binary decision tree, though this is what is taught by Bahl. Instead, the claimed invention relates to preferential processing (such as retention within a limited-capacity cache), with preference being given as a function of hierarchically-related (not binary) assigned priority levels reflected in priority level hint annotations.

Accordingly, there is no teaching or suggestion in Beadle or Bahl of the claimed invention. Further, neither Beadle nor Bahl provides a reason for combining Beadle and Bahl or modifying the teachings of Beadle to arrive at the claimed invention. Further still, the claimed invention is not otherwise rendered obvious by the cited art. For at least these reasons, reconsideration and withdrawal of the rejections of claim 1 are requested respectfully. Claims 3-13 depend from claim 1 and are likewise patentable.

For at least these reasons, reconsideration and withdrawal of the rejection of claims 2-13, 16 and 18-30 are requested respectfully.

B. Beadle, Wu and Kawamoto Fail to Teach Or Suggest Preferential Retention Of Compiled Methods According To Their Respective Indications Of Priority Levels From Among A Hierarchy Of Priority Levels

Independent claim 16 is directed to a method of adapting the interpretation of a p-code file by a virtual machine (VM). The method involves “identifying one or more p-code methods within said p-code file that are annotated with a respective priority indicative annotation indicating a respective priority level from among a hierarchy of priority levels," compiling p-code methods within the p-code file, and "storing said compiled p-code methods in a cache for subsequent execution in place of corresponding interpreted p-code methods, said compiled p-code methods being preferentially retained in said cache in a hierarchical manner corresponding to a hierarchical order of their respective priority indicative annotations."

Thus, claim 16 includes recitations similar to those of claim 1, and is similarly patentable in view of Beadle, Wu and Kawamoto, for reasons similar to those set forth above. The cited discussion of "priority levels" at Kawamoto col. 3, lines 10-46 is unrelated to the claimed invention, as Kawamoto's disclosure relates to priority levels in an entirely different context in which priority is assigned at the time of storage of code in a cache, and which may be changed according to caching logic, as recited at col. 3, lines 10-20 and lines 39-44("when a line is stored in to the cache, if it is stored in accordance with the priority rank assigned to it at this point, this storing approach appears to be adaptive to a dynamic priority change . . . Even if the priority P1 of the line L1 is updated to a lower priority P1' later . . . . The priority is used only when a line is stored into the cache."). This is entirely different from the static priority assigned by annotation of p-code methods for subsequent use in processing/compiling the p-code methods. Claims 18-30 depend from claim 16 and are likewise patentable.

For at least these reasons, reconsideration and withdrawal of the rejection of claims 16 and 18-30 are requested respectfully.

CONCLUSION

In view of the foregoing amendments and remarks, Applicants believe claims 1, 3-13, 16 and 18-30 to be patentable and the application in condition for allowance, and request respectfully issuance of a Notice of Allowance. If any issues remain, the undersigned requests a telephone interview prior to the issuance of an action.

Respectfully submitted,
Jeffrey Wannamaker et al.
by:

Date: September 3, 2008

/Gregory S. Bernabeo/
Gregory S. Bernabeo
Reg. No. 44,032

Saul Ewing LLP
Centre Square West
1500 Market Street, 38th Floor
Philadelphia, PA 19102-2186